

Setup Instructions for Deep Learning

Instructor: Soubhik Barari

1 Overview

Hello! The purpose of this document is to provide directions on technical set-up for this course on Deep Learning. This course centers around the Keras package, a powerful, but accessible high-level framework for neural network programming. We'll be interactively running Python code examples in a series of Jupyter notebooks.

I assume you have intermediate-level Python programming experience and beginner to intermediate-level experience with machine learning. If you consider yourself more of a beginner to machine learning, check out the resources section of this document (last page).

Here are the things you need in order to run the code¹ in this course:

- **Python (version 3.6)** A multi-purpose, object-oriented programming language. This course assumes you have intermediate-level programming experience in Python. We use Python 3, since it is the most up-to-date and ensures that our code will work with all future versions of other packages.
- **Anaconda (version 5.2).** A high-level environment and package manager for Python, optimized for data science. Anaconda includes important libraries used in this course such as `numpy` (numerical computing) and `matplotlib` (visualization).
- **TensorFlow (version 1.10).** A low-level, *super* optimized library developed by Google for building and deploying production-quality neural network models.
- **Keras (version 2.2.2).** A high-level, user-friendly library useful for quickly creating and running neural networks for most social science applications. Uses TensorFlow as its underlying engine.

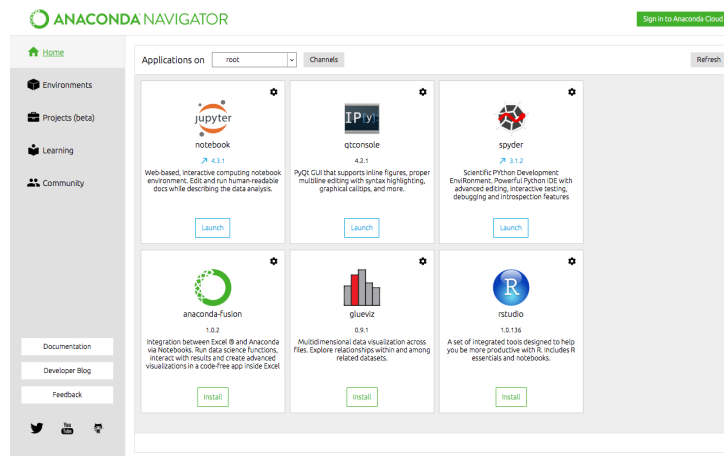
¹ All the coding notebooks for this course can be found in a GitHub code repository at the URL: <http://github.com/soubhikbarari/SAGE-DeepLearning>.

- **Jupyter.** An easy-to-use browser-based notebook for interactively running code, visualizations, and equations. Installing the Anaconda Navigator will automatically install this with the appropriate version.

2 Instructions

Assuming you do not have any of the above components installed, the steps for installation are as follows:

1. **Install Anaconda from its main website.** Visit anaconda.com/download/ and select the appropriate *Python 3.6* installer for your operating system. Download the installer (600MB) and follow the steps to complete installation². Launch the newly installed *Anaconda Navigator* app. It should look like this³:



The Anaconda Navigator is a graphical interface that allows you to use all the different tools, packages, and resources bundled with the Anaconda environment. Feel free to explore!

2. **Install TensorFlow and Keras.** Open up the *Terminal* (*Command Prompt* in Windows) and run the following commands in order to install the libraries not included with Anaconda that we need for our code:

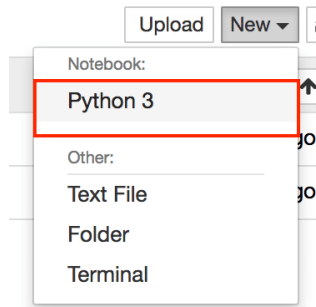
```
conda install tensorflow
```

²In case you get stuck, here are more detailed installation instructions for Windows (docs.anaconda.com/anaconda/install/windows), Mac OS (docs.anaconda.com/anaconda/install/mac-os.html), and Linux (docs.anaconda.com/anaconda/install/linux) respectively.

³(version numbers of the apps will be different)

```
conda install keras
```

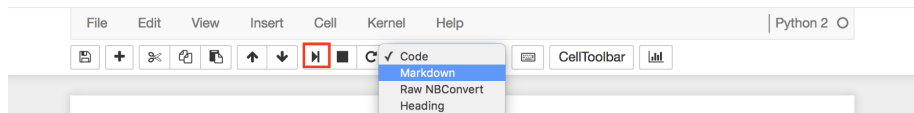
3. **Launch a Python 3 notebook in Jupyter.** Return to the Anaconda Navigator app. Launch the Jupyter application under *Home*. Alternatively, you can launch the Jupyter notebook programmatically using the ‘jupyter notebook’ command on your terminal. After Jupyter opens up in your browser, click on the *New* button and launch a new Python 3 notebook.



4. **Make sure imports work.** Make sure that we’ve correctly installed all the packages by creating a new code block in our notebook (the [+] icon), and typing in the following code:

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
import keras
```

Run this block of code by clicking the “play” button on the Jupyter toolbar:



You should be able to import the above libraries without any errors. You may experience a warning message similar to the following, which you may ignore:

```
/anaconda/lib/python2.7/site-packages/h5py/_init_.py:34: FutureWarning: Conversion of the second
argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated
as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

If you were successfully able to complete all of the steps you are all set to run the coding notebooks in this course!

3 Troubleshooting

Q: 'I have the Python 2.7 version of Anaconda installed. How can I still use Python 3?'

The best way to use Python 3 is to just separately download Anaconda for Python 3, and install the packages and follow the instructions given above.

However, if you'd like to keep your current version of Anaconda (at your own risk!) you can add a notebook *kernel* for Python 3 so that you can run our code notebooks. First, install Python 3.6 (either from python.org/downloads or using the brew command if you're a Mac user). Then, install the Python 3 notebook kernel via the following Terminal commands:

```
python3 -m pip install ipykernel
python3 -m ipykernel install --user
```

You should now be able to create a Python 3 notebook when you launch Jupyter either from the Terminal or through Anaconda.

Q: 'I'd prefer not to use Anaconda. Can't I just use regular ol' Python?'

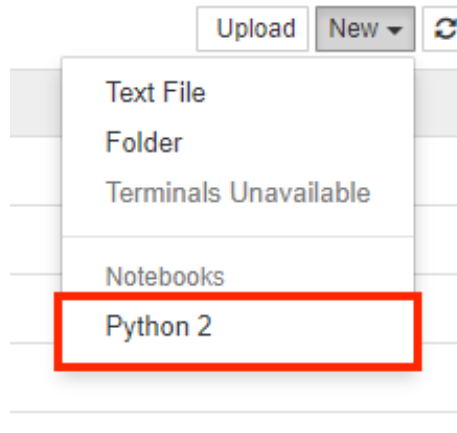
Yes! Just make sure that you install Python 3.6 (either from python.org/downloads or using the brew command if you're a Mac user). With the package manager (Pip) and the command 'pip install [package]' make sure you've installed all the packages and their corresponding versions needed for this course:

- `numpy==1.15.0`
- `matplotlib==2.0.2`
- `tensorflow==1.10.0`
- `keras==2.2.2`
- `jupyter==1.0.0`

Install any additional dependencies as needed.

Q: 'Jupyter works for me, but I can't create a Python 3 notebook. Help!'

When you open Jupyter, this issue might look like this:



This means that Jupyter cannot find a *kernel* for Python 3 in order to create a notebook. You can confirm this via the Terminal command `'jupyter kernelspec list'` which shows all available kernels.

The best way to fix this is to just download Anaconda for Python 3 which includes Jupyter with a Python 3 kernel.

However, if you'd rather not use Anaconda, you can simply install Python 3.6 separately (either from python.org/downloads or using the `brew` command if you're a Mac user), and install Jupyter for Python 3 via the Terminal command `'pip3 install jupyter'`. When you relaunch Jupyter (using the command `'jupyter notebook'`), you should find an option to create a Python 3 notebook.

4 Resources

For details on command line, Python and Jupyter usage I recommend the following resources:

- **Linux/Mac commands:**
<https://www.tjhsst.edu/~dhyatt/superap/unixcmd.html>.
- **DOS (Windows) commands:**
<http://commandwindows.com/command3.htm>.
- **Setting up Jupyter:**
<https://jupyter-notebook-beginner-guide.readthedocs.io>

Given the length of this course, we won't go over each and every line of code in our examples, but it is still useful to know what each block of code is doing.

Thus, if you are relatively new to machine learning or would like a refresher on machine learning coding conventions in Python, I would recommend getting some

basic familiarity with the `scikit-learn` library⁴. `scikit-learn` is the most popular machine learning in Python and its syntax has established many of the coding conventions and norms (e.g. dataset loading, variable names, object-centric operations, plotting) in machine learning. Additionally, we will be using the `matplotlib` visualization library⁵ which is commonly used to plot datasets, outputs, and model characteristics in machine learning.

Here are some helpful examples that practically demonstrate how to do different machine learning tasks using `scikit-learn` (deep learning code is not much different!):

- **Fitting machine learning algorithms to data:**
http://scikit-learn.org/stable/tutorial/statistical_inference/settings.html#datasets
- **Basic end-to-end machine learning pipeline:**
http://scikit-learn.org/stable/tutorial/statistical_inference/putting_together.html
- **Plotting predictions from a regression model:**
http://scikit-learn.org/stable/auto_examples/plot_cv_predict.html
- **Plotting images using matplotlib:**
https://matplotlib.org/gallery/images_contours_and_fields/image_demo.html
- **Classifying topics in news documents (step-by-step explanation):**
http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

⁴Scikit-learn website: <http://scikit-learn.org/>

⁵Matplotlib website: <http://matplotlib.org/>